Exploiting Remote Memory Operations to Design Efficient Reconfiguration for Shared Data-Centers over InfiniBand

P. Balaji, K. Vaidyanathan, S. Narravula, K. Savitha, H. –W. Jin
D. K. Panda
Network Based Computing Laboratory
The Ohio State University
COTS Clusters

• Advent of High Performance Networks
  – Ex: InfiniBand, Myrinet, Quadrics, 10-Gigabit Ethernet
  – High Performance Protocols: VAPI / IBAL, GM, EMP
  – Provide applications direct and protected access to the network

• Commodity-Off-the-Shelf (COTS) Clusters
  – Enabled through High Performance Networks
  – Built of commodity components
  – High Performance-to-Cost Ratio
InfiniBand Architecture Overview

- Industry Standard
- Interconnect for connecting compute and I/O nodes
- Provides High Performance
  - Low latency of lesser than 4us
  - Over 935MBps uni-directional bandwidth
  - Offloaded Transport Layer; Zero-Copy data-transfer
  - Provides one-sided communication (RDMA, Remote Atomics)
- Becoming increasingly popular
Cluster-based Data-Centers

• Increasing adoption of Internet
  – Primary means of electronic interaction
  – Highly Scalable and Available Web-Servers: Critical!

• Utilizing Clusters for Data-Center environments?
  – Studied and Proposed by the Industry and Research communities

• Nodes are logically partitioned
  – Interact depending on the query
  – Provide services requested
  – Services provided are related
  – Fragmentation of resources

(Courtesy CSP Architecture Design)
Shared Multi-Tier Data-Centers

Hosting several unrelated services on a single clustered data-center
Issues in Shared Data-Centers

• Hosting several unrelated services on a single data-center
  – Ex: A single data-center hosting multiple websites
  – Currently used by several ISPs and Web Service Providers (IBM, HP)
  – Allows differentiation in resources provided for each service
  – Fragmentation is a big concern!

• Over-provisioning of nodes for each service
  – Nodes provided to each service based on the worst-case estimates
  – Widely used approach
  – Leads to severe under-utilization of resources
Dynamic Reconfigurability

Nodes reconfigure themselves to highly loaded websites at run-time
Objective

• Under Utilization of resources needs to be curbed

• Dynamically Configuring nodes allotted to each service
  – Widely studied approach for Clusters
  – Interesting Challenges in the Data-Center Environment
    • Highly loaded back-end servers
    • Compatibility with existing applications (Apache, MySQL, etc)

• Can the advanced features provided by InfiniBand help?
Presentation Roadmap

- Introduction and Background
- Shared Data-Centers
- Designing Dynamic Reconfigurability for Shared Data-Centers
- Experimental Results
- Concluding Remarks
- Continuing and Future Work
Shared Data-Centers Overview

- Clients request services using high level protocols such as HTTP
- Requests are distributed to the nodes using load-balancers
  - Load Balancers expose a single IP address to the clients
  - Maintain a list of several internal IP addresses to forward the requests
- Several solutions for load-balancers
  - Hardware Load-Balancers
  - Software Load-Balancers
  - Cluster-based load-balancers
Cluster-based Load Balancers

• Hardware Load-Balancers
  – Commonly used in several environments
  – In-flexible and cannot be tuned to the data-center requirements

• Software Load-Balancers
  – Easy to modify and tune to the data-center requirements
  – Potential bottlenecks for highly loaded data-center environments

• Cluster-based load-balancers
  – Proposed by several researchers as an additional Edge Tier [shah01]
  – Provides intelligent services such as load-balancing, caching, etc
  – Use an additional hardware load-balancer or DNS aliasing to get requests

Presentation Roadmap

- Introduction and Background
- Shared Data-Centers
- Designing Dynamic Reconfigurability for Shared Data-Centers
- Experimental Results
- Concluding Remarks
- Continuing and Future Work
Design Issues

- Support for Existing Applications
  - Modifying existing applications: Cumbersome and Impractical
  - Utilizing External Helper Modules (external programs running on each node)
    - Take care of load monitoring, reconfiguration, etc.
    - Reflect changes to the data-center applications using environment settings

- Load-Balancer based vs. Server based Reconfiguration
  - Trading network traffic for CPU overhead
  - Load Balancers “convert” nodes to serve their website

- Remote Memory Operations based Design
  - Server node applications are typically very compute intensive
  - Execution of CGI scripts, business logic, database processing
  - Utilizing one-sided operations provided by InfiniBand
  - Load-balancers remotely monitor and reconfigure the system
Implementation Details

- History Aware Reconfiguration
  - Avoiding Server Thrashing by maintaining a history of the load pattern

- Reconfigurability Module Sensitivity
  - Time Interval between two consecutive checks

- Maintaining a System Wide Shared State

- Shared State with Concurrency Control

- Tackling Load-Balancing Delays
System Wide Shared State

- Nodes in the cluster need to share control information
  - Load, Current State of the node, etc.

- Sockets based Implementation has several disadvantages
  - All communication needs to be explicitly performed
  - Asynchronous requests need to be handled by the host
    - A major concern due to the high CPU overhead on the servers

- InfiniBand RDMA operations try to avoid these disadvantages
  - Load-balancers can share data on the servers using RDMA Read
  - Can update system state using RDMA Write and Atomic Operations
Shared State with Concurrency Control

• Load-balancers query the system load at regular intervals
• On detecting a high load, a reconfiguration is done
• Multiple Concurrency issues to be dealt with:
  – Multiple simultaneous transitions possible
    • Each node in the load-balancer cluster can attempt a reconfiguration
    • Multiple nodes might end up being converted on a single burst
  – Hot Spot Effects on remote nodes
    • All load-balancers might try to get load information from the same node
    • They might try to convert the same node
  – Additional Logic Required!
Locking Mechanism

- We propose a two-level hierarchical locking mechanism
  - Internal Lock for each web-site cluster
    - Only one load-balancer in a cluster can attempt a reconfiguration
  - External Lock for performing reconfiguration
    - Only one web-site can convert any given node
  - Both locks performed remotely using InfiniBand Atomic Operations
Tackling Load-Balancing Delays

- Load-Balancing Delays
  - After a reconfiguration, balancing of load might take some time
  - Locking mechanisms only ensure no simultaneous transitions
  - We need to ensure that all load-balancers are aware of reconfigurations

- Dual Counters
  - Shared Update Counter (SUC)
  - Local Update Counter (LUC)

- On reconfiguration:
  - LUC should be equal to SUC
  - All remote SUCs are incremented
Presentation Roadmap

1. Introduction and Background
2. Shared Data-Centers
3. Designing Dynamic Reconfigurability for Shared Data-Centers
4. Experimental Results
5. Concluding Remarks
6. Continuing and Future Work
Experimental Test-bed

• Cluster 1 with:
  – 8 SuperMicro SUPER X5DL8-GG nodes; Dual Intel Xeon 3.0 GHz processors
  – 512 KB L2 cache, 1 GB memory; PCI-X 64-bit 133 MHz

• Cluster 2 with:
  – 8 SuperMicro SUPER P4DL6 nodes; Dual Intel Xeon 2.4 GHz processors
  – 512 KB L2 cache, 512 MB memory; PCI-X 64-bit 133 MHz

• Mellanox MT23108 Dual Port 4x HCAs; MT43132 24-port switch

• Apache 2.0.50 Web and PHP servers; MySQL Database server

• Experimental Results (Outline)
  – Basic IBA Performance
  – Impact of Background Computation Threads
  – Impact of Request Burst Length
  – Node Utilizations
Basic IBA Performance

- RDMA Read operation on IBA outperforms TCP/IP (IPoIB)
  - IBA achieves about 12us latency compared to the 56us of IPoIB
  - IBA achieves about 830 MBps bandwidth compared to the 230 MBps of IPoIB
  - More importantly near zero CPU requirements on the receiver side
Impact of Background Threads

Impact on Latency

Impact on Bandwidth

• Remote memory operations are not affected AT ALL with remote server load
• Ideal for the data-center environment
Impact of Burst Length

- Rigid has 3 nodes for each website; Over-provisioning has 6 nodes for each website
- Large Burst Length allows reconfiguration of the system closer to the best case!
- Performs comparably with the static scheme for small burst sizes
Node Utilization for 3 Co-hosted Web sites

For Burst Length = 512

For Burst Length = 8096

- For large burst lengths, the reconfiguration time is negligible; performance is better
Presentation Roadmap

1. Introduction and Background
2. Shared Data-Centers
3. Designing Dynamic Reconfigurability for Shared Data-Centers
4. Experimental Results
5. Concluding Remarks
6. Continuing and Future Work
Concluding Remarks

• Growing Fragmentation of resources in data-centers
  – Related services provided by Multi-Tier Data-Centers
  – Unrelated services provided by Shared Data-Centers

• Dynamically configuring resources allotted
  – A common approach used in clusters
  – Data-Center environment has its own challenges
    • Highly loaded back-end servers
    • Compatibility with existing applications

• Provided a novel approach utilizing the RDMA features of IBA
  – A scheme resilient to the load on the back-end servers
  – Demonstrated up to 2.5 times improvement in the throughput
  – Similar performance using only half the nodes
Presentation Roadmap

- Introduction and Background
- Shared Data-Centers
- Designing Dynamic Reconfigurability for Shared Data-Centers
- Experimental Results
- Concluding Remarks
- Continuing and Future Work
Continuing and Future Work

- Multi-Stage Reconfigurations
  - Least loaded servers might not be the best server to reconfigure
  - Caching constraints
  - Replicated Databases
  - Hardware heterogeneity

- Utilizing Dynamic Reconfigurability for advanced services
  - QoS guarantees
  - Differentiation in the resources provided
Thank You!

For more information, please visit the NBC Home Page

http://nowlab.cis.ohio-state.edu

Network Based Computing Laboratory,
The Ohio State University
Backup Slides